Al-Monakkeb - A Superior Indexer for Arabic Texts

Ramzi A. Haraty and Samer A. Khatib
Lebanese American University
Email: rharaty@lau.edu.lb; khatibsam@hotmail.com

ABSTRACT

Stemming has a large effect on Arabic information indexing and retrieval, at least partially due to the highly inflected nature of the language. Our work demonstrates the process of improving the stemmer of Daher, [2]. We reached a recall difference of 28%. The main part of improvement was due to the addition of more grammatical rules that facilitate the process of stemming.

Keywords: Arabic text, automatic indexing, and stemming.

1. INTRODUCTION

Up till recently, the work on the stemming and indexing of the Arabic documents followed by the creation of the subject heading, was not satisfactory. The reason for this dissatisfaction was due to bad results in the stemming process. The stemming process as we know is the backbone of the indexer and the subject heading creation. The recall[1] of the stemmer was not reaching even 65% [3]. Since Arabic is the official language of over twenty Middle Eastern and African countries, it is not acceptable to have automatic indexers, subject heading creation with a low recall stemmer. Working on these topics will have a great effect on our society and education level because it will make the path to correct data shorter.

In this paper we work on the Arabic Auto-Indexing, we achieve a significant improvement primarily on the work of [2], and others. Our main field of improvement was on the stemmer: we got a recall result of 75% compared to their result of 46%.

The rest of the paper is organized as follows: section 2 presents related work in the filed of stemming and indexing. In section 3, we describe our stemmer and the improvements done on it and the grammatical rules that were added. Also, we show the code that describes those rules. In section 4, we present the experimental results. The results show the recall and precision of both our work and the work of [2] for each and every text and an over all result. A conclusion is drawn in section 5.

2. RELATED WORK

In Arabic automatic-indexing, our main work and improvement is in the stemming module. Thus, our main emphasis will be on the previous work done on Arabic stemming. Four different approaches to Arabic stemming can be identified [4]:

a. Manually constructed dictionaries.
b. Algorithmic light stemmers, which remove prefixes and suffixes.
c. Morphological analyses, which attempt to find roots.
d. Statistical stemmers, which group word variants using clustering techniques.

Manually constructed dictionaries [5] built up dictionaries of roots and stems for the words to be indexed. [6] develop a set of lexicons of Arabic stems, prefixes, and suffixes, with truth tables indicating legal combinations. This type of stemming is a table-based. This type of stemming is also used in [7], where they proposed a novel thesaurus-based technique.

Light stemmers [8] remove the suffixes and prefixes from words, without trying to deal with infixes, or recognize patterns and find roots. In [9] the authors present two stemming algorithms for Arabic information retrieval systems: the root-based stemmer and the light stemmer. The aim of this technique is not to produce the linguistic root of a given Arabic surface form; it is rather to remove the most frequent suffixes and prefixes. Some used the light stemming in their work as in [11]. [12], [13] and [14] used also light stemmers in their work. [15] tested three types of light stemmers: the Al-stem, the U Mass stemmer and the modified U Mass stemmer. The result shows that the modified U Mass stemmer did achieve mean average precision results that were statistically better than the two other stemmers.

Stemmers that use morphological analysis to stem words and get their roots are more advanced than the previous two types of stemming. [10] is one example of this type of stemmers where the affixes[2] are removed from words to get the root.

[1] Recall is the percentage of the number of words that are retrieved relevant to the number of all correct stem word that should be retrieved by an indexer.

[2] Prefixes, infixes, and suffixes.

Statistical stemmers group word variants using clustering techniques. [11] developed a clustering algorithm for Arabic words having the same verbal root. They used root-based clusters to substitute for dictionaries in indexing for information retrieval.

In this work we focus on the third type, which is the morphological analysis to retrieve the root of the words. This does not mean that our work does not belong to the first or second types. However, we use some tables that contain the words and their roots (e.g., the name of the countries or the temporal stop lists that make our work belong to the first type) and we definitely remove the suffixes and prefixes, which make our work belong to the second type.

3. THE STEMMER

Stemming is the process of retrieving the root of words. This process of retrieving roots is done by passing the word by a number of routines. Those routines are special conditions that try to: remove the additions[3] to the word, find the rhythm of the words, and restructure the word to ease the root retrieval process. [2] has worked on those routines, but due to the complexity of Arabic language, not all the conditions and rules were done. We continued what the process [2] had started and added many improvements to this work. Below are some of these improvements:

A. The Alif Problem

In Arabic we have different representations of the Alif. We have the “أ” which is pronounced as “Aa” , “إ” which is pronounced as “Ii” , “آ” which is pronounced as “Aaaa” and the normal “ا” which is pronounced as “a”. The stemming problems start when we face a word like “ألأول” which is stemmed to “ألأول”. However, it should be stemmed to “اول”. To solve this problem we replace all the (“أ”, “إ”, and “آ”) with “ا”.

B. The End Letters Problems

Some words like “مدرستي” and “مدرستى” (“my school”) differ only by the two letters “ي” and “ى”, and they both should be stemmed to one word “درس”. [2] stems the first word to “درست”, and the second is unchanged and stemmed to itself. To solve this problem, we replace all the final “ى” with “ي”.

The second end letters problem is the “ه” and “ة” pronounced “haa” and “taa”. In [2] some words do not face problem like “مدرسة” and “مدرسه” (“school”); they are both stemmed to “درس” (“study”). But others like “فاحه” and “فاحة” (“the smell diffused”) are stemmed to different words: “فاح” and “فحة”. To solve this problem as in the “ي” and “ى”, we replace all the final “ة “ with “ه”.

Also the word “الى”, which is used so much in the Arabic language, has a wrong stemming, which is “ى”, and the above procedure was able to solve this problem as well.

C. The Letter of Atef or Conjunction or other First Additions

The letters “و” and “ف” at the beginning of some words are most likely to be not part of the words’ conjunction letters. This problem is due to the way people write the words. For example, this sentence is written in two ways: “الطفل يأكل و يشرب” and “الطفل يأكل ويشرب” (“the child eats and drinks”). In the first was the letter “و” is a stand alone letter, while in the second it is part of the word “يشرب” (“drink”). The word “ويشرب” will stay the same and be stemmed to “ويشرب” in algorithm of [2], while the correct stemming is “شرب”. This case was handled in two steps (see algorithm below): (1) we run the stemmer for the first time to be sure that these letters “و” and “ف” are not a part of the word; if we do not reach a suitable root, (2) we remove those two letters from the word and run the stemming the second time:

```
SomeChar = CharAt(SomeWord, 1)
If SomeChar = "و" Or SomeChar = "ف" Or
SomeChar = "ا" Or SomeChar = "ب" Or
SomeChar = "ت" Or SomeChar = "ل" Then
SomeWord = Mid(SomeWord, 2)
GoTo TopOfPro
End If
```

The presence of the “Or SomeChar = "ا"” and the rest of “Or” conditions are simply to solve problems we faced in stemming words as “اتكذبين” or “اتعلمي” that were stemmed to “اتكذب” “اتعلم”or “ل” to solve the problem of “لاصدار” that was stemmed to “لاصدار”.

D. Harakat or Tanwin and Punctuation

Arabic has “harakat” (short vowels) that have the ability to completely change the meaning of the word; for example, “قِمة” and “قُمة”. The first word means top of the mountain or peak, and the second means garbage. These words have the same spelling, but completely different meaning, just because of the use of “ُ◌” and “ِ◌” . We added a table in the database containing all the harakat; and every word before being stemmed passes through special routines that remove all the harakat from it. The same procedure is also responsible for removing the punctuation.

[3] Suffix, prefix, tatweel, etc.

E. Kashida or Tatweel

Arabic is the only language that has the characteristic of the tatweel. One can add to a word the character “-” for aesthetic reasons. A small example is the word “معلمة” (“teacher”) that could also look like “مـــعلمـــة”. This has nothing to do with the meaning; only the format of the word is affected. Thus, before stemming the word, all those additional characters should be removed; and this is done in the same procedure that removes the punctuation.

F. The Letter “ل”

In the future tense, [2] has treated the “س” (“S”) letter at the beginning of the sentences. The letter “ل” (“L”) at the beginning of the sentence is not used as future tense, but programming-wise it is treated as “س” (“S”) letter because it is added to the present tense and at the beginning of the sentence. We amended the stemmer to handle such cases.

G. Deciding between a Verb and a Noun

Deciding between a verb and a noun is an important step in stemming. Any wrong decision will most likely lead to bad consequences. The following amendments were added to the stemmer to enhance the decision making process:

i. Verb Problem

```
ElseIf WordsRhyme("فعل", ThisWord) Then
TempType = TYPE_VERB
```

The above two lines were removed because any word of the rhythm “فعل” will be treated as verb. For example, if the sentence “عين تالم” is stemmed in [2], the word “عين” will be treated as “verb” and the word “تالم” will be treated as “unknown”. However, in our stemmer the word “عين” will be treated as “unknown” and the word “تالم” will be also “unknown”; but later in the stemmer, this word will be given a “verb” type.

ii. Removal of “Noun” Type:

A very important change in the stemmer is the removal of the type “Noun” and the inclusion of all the types that are different from verb as “Unknown”. The reason for this change is due to the fact that, in stemming, the meaning of the word does not change if it is an adjective or a noun; however, the meaning differs if it is a noun or a verb.

H. Expected Word Type

Daher [2] used five conditions and eleven stop list terms in order to predict the type of the following or next word. We have raised these conditions to 17, and the terms that were able to predict the following word type were raised from 11 to 95 conditions. The conditions were able to predict the type of the following word:

1. SL_NASB [1] “ادوات النصب” are always followed by a verb. Words like “لن-كي-لكي” (“in order to” - “for” - “will not”) are examples of such words. Other “ ادوات نصب” like “اذن – أن – فاء السببية” follow different rules and are not considered.

2. SL_JASM “احرف الجزم” are always followed by a verb, like “لم-لا-سوف-لما”. However, words such as “إنْ” are not taken into consideration because they can be written in different ways and will lead to a conflict in deciding the type.

3. SL_JAR “احرف الجر” are always followed by a noun. However, we took only six of them “من-في-ففي-عن-الى-على”. The rest, like “ك-الباء-رُب-...“, are not considered because they have multi-usage purposes (e.g., “رُب” (“likely”) that is followed by a noun and “رب” (“God”) that could be followed by different types). When these pass in the new stemmer that removes the “Harakat” they will be the same word.

4. SL_ZAREF “ظرف الزمان والمكان”, like “- تحت-لدي- - دون بين- لدن-فوق”, are always followed by a noun.

5. SL_MONADA “احرف النداء”, like “يا-ايا-هيا-وا” followed by a noun.

6. SL_SHART “ادوات شرط”, like “مهما-اينما-حيثما-كيفما” are followed by a verb. Some words like “أنَّى”, are also “أداة شرط”, but after removing the Harakat, they can have different meanings.

I. Words with the Same Stemming but Having Different Meanings

Some words like “الذهب” and “يذهب” or “الملعب” and “يلعب” are stemmed to the same word “ذهب” or “لعب”. However, they both can have a different meaning according to their position in the sentence and the way they are used. The first word can be a noun that means “gold”, and the second can be a verb that means “go”. The problem was solved by adding a field to the list of stemmed words to clarify whether this stemmed word is a verb, a noun, or other. Thus, the sentence “ احضر الذهب الخام ولكنه لم يذهب به الى الصائغ” (“he got the unpurified gold but did not take it to the goldsmith”), when stemmed by [2], the result of “الذهب” and “يذهب” was the root “ذهب”, which is totally wrong. In our stemmer, the result of “الذهب” and

“يذهب” will be two words “ذهب” with two different types: noun and verb, and the count of each will be one.

4. EXPERIMENTAL RESULTS

In this section, we describe the process of testing the automatic indexer. We used 25 different texts unaltered, that is, without touching the structure of the text. For example, we did not change the letter “و” (“and”) from the original word, (e.g. “والطابة”) (“and the ball”) to “ و الطابة” (i.e., creating a space between “و” and “الطابة”). We also kept the Harakat and the tatweel. The texts were of different lengths. We denoted the number of words of the text by “L”. We also denoted by “N” the number of the words that need to be stemmed, and these represent every word in the text except the stop list words and duplicated words. Our method will retrieve RR (retrieved) relevant index words, RI (retrieved) irrelevant index words, and NRR (not retrieved) relevant words; that is, NRR = N – RR.

To evaluate the results, we include the two usual metrics: precision and recall. Precision measures the percentage of relevant results; i.e., how well the retrieval algorithm avoids returning results that are not relevant. In other words, precision = RR / (RR + RI). Recall measures the completeness of retrieval of relevant indices. That is Recall = RR / (RR + NRR) = RR / N.

Table 1: The results of stemming the 25 text.

Text#	L	N	RR	RI	NRR	Recall	Precision
1	467	192	134	3	58	0.70	0.98
2	1044	543	427	1	116	0.79	1.00
3	789	392	292	5	100	0.74	0.98
4	532	261	194	3	67	0.74	0.98
5	532	298	230	7	68	0.77	0.97
6	546	348	261	5	87	0.75	0.98
7	587	364	267	5	97	0.73	0.98
8	887	471	363	7	108	0.77	0.98
9	1039	559	429	6	130	0.77	0.99
10	846	416	291	3	125	0.70	0.99
11	1023	557	393	7	164	0.71	0.98
12	714	417	319	7	98	0.76	0.98
13	767	384	281	7	103	0.73	0.98
14	513	304	232	10	72	0.76	0.96
15	413	225	167	3	58	0.74	0.98
16	424	239	188	4	51	0.79	0.98
17	463	259	191	6	68	0.74	0.97
18	1231	700	524	3	176	0.75	0.99
19	577	358	261	6	97	0.73	0.98
20	559	282	199	3	83	0.71	0.99
21	496	297	234	2	63	0.79	0.99
22	451	237	188	3	49	0.79	0.98
23	599	345	249	9	96	0.72	0.97
24	633	346	244	6	102	0.71	0.98
25	720	331	243	2	88	0.73	0.99
Average			**272.04**	**4.92**	**92.96**	**0.74**	**0.98**

Table 1 shows a recall of 74% and a precision of 98%, which, we think is a very good result for a complicated language as Arabic. These results prove to be of great use for people who construct indexes for Arabic documents. [2] used the first 24 texts for testing. However, the authors selected index words by setting the weight threshold to two, which also decreases the number of words to be stemmed (N), hence increasing the recall percentage. They showed a result on average recall of 46% and precision of 64% (see Table 2).

Table 2: Comparison between our and [2] results.

		Our Result		**[19] Result**			
Text#	L	Recall	Precision	**Recall**	**Precision**	**Diff-R**	**Diff-P**
1	467	0.70	0.98	0.49	0.53	0.21	0.45
2	1044	0.79	1.00	0.49	0.73	0.30	0.27
3	789	0.74	0.98	0.49	0.68	0.25	0.30
4	532	0.74	0.98	0.48	0.63	0.26	0.35
5	532	0.77	0.97	0.53	0.61	0.24	0.36
6	546	0.75	0.98	0.43	0.58	0.32	0.40
7	587	0.73	0.98	0.4	0.56	0.33	0.42
8	887	0.77	0.98	0.35	0.68	0.42	0.30
9	1039	0.77	0.99	0.47	0.69	0.30	0.30
10	846	0.70	0.99	0.44	0.67	0.26	0.32
11	1023	0.71	0.98	0.4	0.6	0.31	0.38
12	714	0.76	0.98	0.59	0.77	0.17	0.21
13	767	0.73	0.98	0.49	0.69	0.24	0.29
14	513	0.76	0.96	0.51	0.67	0.25	0.29
15	413	0.74	0.98	0.53	0.6	0.21	0.38
16	424	0.79	0.98	0.54	0.58	0.25	0.40
17	463	0.74	0.97	0.46	0.66	0.28	0.31
18	577	0.73	0.98	0.45	0.71	0.28	0.27
19	559	0.71	0.99	0.41	0.56	0.30	0.43
20	496	0.79	0.99	0.53	0.62	0.26	0.37
21	451	0.79	0.98	0.47	0.64	0.32	0.34
22	599	0.72	0.97	0.32	0.58	0.40	0.39
23	633	0.71	0.98	0.43	0.58	0.28	0.40
24	720	0.73	0.99	0.37	0.54	0.36	0.45
Average		**0.75**	**0.98**	**0.46**	**0.64**	**0.28**	**0.35**

We also conducted another experiment by comparing the result of our stemmer and [2] on separate texts without using any threshold. The results are as follows:

1. Total number of words of the text = 720.
2. Number of words that should be stemmed without removing duplicates = 484.
3. Number of words correctly stemmed by our improved stemmer = 366.
4. Number of words correctly stemmed by [2] = 198.
5. Words that should not be stemmed, but stemmed wrongly by [2] = 34.
6. Words that should not be stemmed, but stemmed identically by [2] =7.

While the recall of [2] is 40.9%, our recall is 76.6%, recording a great improvement. We also compared our results with [10]. [10] achieved an accuracy of 96% but mentioned nothing about recall. However our precision is 98%. [3] also claimed achieving an accuracy of 97%, using a dictionary of 4,748 trilateral and quadrilateral roots while our root stemming was without the help of any dictionary.

5. CONCLUSION

Stemming has a large effect on Arabic information indexing and retrieval, at least partially due to the highly inflected nature of the language. Our work demonstrates the process of improving the stemmer of [2] .We reached a recall difference of 28%. The main part of improvement was due to the addition of more grammatical rules that facilitated the process of stemming. The big list of stop words also helped in predicting the type of the coming word. We also included a list of all the names of countries that prevented the wrong stemming of these words. Assigning the correct type of the stemmed words was also a part of our improvement process, because we distinguished between the same stemmed words with different types, like: "ذهب" ("went") and "ذهب" ("gold"). The problems of Harakat, Tanwiin, Punctuation and Tatweel were also solved to facilitate the stemming process.

Further research should improve the stemming process of the auto-indexing system. Adding more grammatical rules and conditions and performing more analysis to the word before sending it to the stemmer, should increase the percentage of both the recall and the precision

REFERENCES

[1] Sharaf, M. Y. 1990. Sharh Katr al Nada wa Bal al Sada li Ibn Hisham (1309-1360). Librairie du Liban Publishers, Beirut, Lebanon (in Arabic). p. 352.

[2] Mansour, N., Haraty, R., Daher, W., and Houri, M. 2005. *An auto-indexing method for Arabic text*. Special Issue of the Journal of Computational Methods in Sciences and Engineering (JCMSE): Intelligent Systems.

[3] Khoja, S. 2001. *APT: Arabic Part-of-speech Tagger*. URL:http://archimedes.fas.harvard.edu/mdh/arabic/NAACL.pdf [10 March 2003].

[4] L. Larkey, L. Ballesteros, and M.E. Connell. 2002. *Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis*. In *SIGIR'02, August 11-15, 2002, Tampere, Finlan*d. pages 275–282.

[5] Al-Kharashi, I., and Evens, M. 1994. Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System. Journal of the American Society for Information Science. **45**(8): 548-560.

[6] *Buckwalter T.* URL: http://www.qamus.org/morphology.htm [7 July 2003].

[7] Xu, J., Fraser, A., and Weischedel, R. 2002. Empirical studies in strategies for Arabic retrieval. In: *Annual ACM Conference on Research and Development in Information Retrieval.* Tampere, Finland, pages: 269 – 274. ACM Press New York, NY, USA.

[8] Chen, A., and Gey, F. 2002. Building an Arabic Stemmer for Information Retrieval. In: *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, Nov 19-22. Gaithersburg, Maryland USA, pages 269-274. National Institute of Standards and Technology, Gaithersburg, Maryland, USA.

[9] Aljlayl, M., and Frieder, O. 2002. On Arabic search: improving the retrieval effectiveness via a light stemming approach. In: *Proceedings of the eleventh international conference on Information and knowledge management*, McLean, Virginia, USA, pages: 340-347. ACM Press, New York, NY, USA.

[10] Khoja, S., and Garside, R. 1999. *Stemming Arabic tex*t. URL: http://www.comp.lancs.ac.uk/computing/users/khoja/stem-mer.ps [1 November 2003]

[11] De Roeck, A. N., and Al-Fares, W. 2000. A morphologically sensitive clustering algorithm for identifying Arabic roots. In Proceedings *of The 38th Annual Meeting of the Association for Computational Linguistics ACL-2000,* October 1-8, 2000, Hong Kong.

[12] Aljlayl, M., Beitzel, S., Jensen, E., Chowdhury, A., Holmes, D., Lee, M., Grossman, D., and Frieder, O. 2001. IIT at TREC-10. In: *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, Nov 13-16. Gaithersburg, Maryland USA, pages 265-274. National Institute of Standards and Technology, Gaithersburg, Maryland, USA.

[13] Savoy, J., and Rasolofo, Y. 2002. Report on the TREC-11 Experiment:
Arabic, Named Page and Topic Distillation Searches. In: *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, Nov 19-22. Gaithersburg, Maryland USA. National Institute of Standards and Technology, Gaithersburg, Maryland, USA.

[14] Xu, J., Fraser, A., and Weischedel, R. 2001. "TREC 2001 Cross-lingual Retrieval at BBN". In: *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, Nov 13-16. Gaithersburg, Maryland USA, page 68. National Institute of Standards and Technology, Gaithersburg, Maryland, USA

[15] Darwish, K., and Oard, D. 2002. Evidence Combination for Arabic-English Retrieval. In: *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, November 19-22. Maryland USA.